

## AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A ~~classfile modification method, method for modifying a software application comprising:~~  
~~modifying inserting functions calls at entry points and exit points of each method associated with the software application via a bytecode modifier;~~  
~~modifying a classfile after said classfile has been compiled from a source code version of the software application, said classfile describing properties of a class within an object oriented environment;~~ ~~said modifying comprising:~~  
modifying a method information structure ~~for each method associated with the software application by adding byte code instructions to the byte code instructions of said method information structure~~ ~~structure's respective method, said byte code instructions to cause a plug-in module's handler method associated with a plug-in handler to execute an output function for said each method, the plug-in handler to record method information associated with methods at each entry point and exit point;~~  
~~adding a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions;~~  
compiling results of the modifying of the classfile, the results including  
method information, ~~the method information including a~~

~~dependency hierarchical tree indicating dependency order of the methods, and a time hierarchical tree indicating chronological order of the methods; and~~

~~filtering the method information<sub>1</sub> by applying filtering parameters via a filtering module, according to user preferences and the dependency and time hierarchical trees~~  
~~the filtering of the method information including filtering timing data, method invocations, and other method-related information<sub>1</sub>.~~

2. (Cancelled) .
3. (Currently Amended) The ~~classfile modification method~~ of claim 2 wherein said ~~modifying a classfile further comprises~~ further comprising:  
adding a field information structure to the methods, said field information structure describing a field that is to store a numeric identifier of said class.
4. (Currently Amended) The ~~classfile modification~~ method of claim 3 wherein said numeric identifier is provided to said class by a method of which said ~~a~~ dispatch unit is comprised.
5. (Currently Amended) The ~~classfile modification~~ method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said ~~an~~ output function treatment in response to an entry point of said method being reached.
6. (Currently Amended) The ~~classfile modification~~ method of claim 5 wherein said output function treatment is a function selected from the group consisting of:
  - 1) recording a time of entry for said method;

- 2) recording an input parameter value for said method; and,
  - 3) incrementing a counter for said method.
7. (Currently Amended) The ~~classfile modification~~ method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to an exit point of said method being inevitably reached.
8. (Currently Amended) The ~~classfile modification~~ method of claim 7 wherein said output function treatment is a function selected from the group consisting of:
- 1) recording a time of entry for said method;
  - 2) recording an input parameter value for said method; and,
  - 3) incrementing a counter for said method.
9. (Currently Amended) The ~~classfile modification~~ method of claim 7 wherein the portions of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to any exit point of said method being inevitably reached.
10. (Currently Amended) The ~~classfile modification~~ method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising during execution of said method.
- 11-12 (Cancelled)
13. (Currently Amended) The ~~classfile modification~~ method of claim ~~12~~1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokestatic instruction.
14. (Currently Amended) The ~~classfile modification~~ method of claim ~~12~~1 wherein said byte code instructions are Java compatible and wherein said at least one of

said instructions is an invokevirtual instruction.

15. (Currently Amended) The ~~classfile modification~~ method of claim ~~12~~ wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokespecial instruction.

16-21 (Cancelled)

22. (Currently Amended) A machine readable storage medium ~~containing~~ comprising instructions to modify a software application which, when executed by a machine, ~~cause a classfile modification method to be performed; said classfile modification method comprising machine to:~~ modifying inserting functions calls at entry points and exit points of each method associated with the software application via a bytecode modifier; ~~modifying a classfile after said classfile has been compiled from a source code version of the software application, said classfile describing properties of a class within an object oriented environment; said modifying comprising:~~ modifying modify a method information structure for each method associated with the software application by adding byte code instructions to the byte code instructions of said method information structure's respective method; ~~said byte code instructions structure to cause a plug-in module's handler method associated with a plug-in handler to execute an output function for said each method, the plug-in handler to record method information associated with methods at each entry point and exit point;~~ adding a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module

~~during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions;~~  
~~compiling compile~~ results of the modifying of the classfile, the results including  
~~method information, the method information including a dependency hierarchical tree indicating dependency order of the methods, and a time hierarchical tree indicating chronological order of the methods; and~~  
~~filtering filter~~ the method information, by applying filtering parameters via a  
~~filtering module, according to user preferences and the dependency and time hierarchical trees~~the filtering of the method information including  
~~filtering timing data, method invocations, and other method-related information;~~

23. (Cancelled)
24. (Currently Amended) The machine readable storage medium of claim 23 wherein  
~~said modifying a classfile further comprises the instructions which, when executed, further cause the machine to:~~  
~~adding add~~ a field information structure ~~to the methods~~, said field information structure describing a field that is to store a numeric identifier of said class.
25. (Cancelled)
26. (Currently Amended) The machine readable storage medium of claim 22 wherein  
a portion of said byte code instructions that are added to said method are for  
causing said plug-in ~~module's~~ handler method to provide said output function  
treatment in response to an entry point of said method being reached.
27. (Cancelled)

28. (Currently Amended) The machine readable storage medium of claim 22 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to an exit point of said method being inevitably reached.
29. (Cancelled)
30. (Currently Amended) The machine readable storage medium of claim 28 wherein ~~the~~ portions of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to any exit point of said method being inevitably reached.
31. (Currently Amended) The machine readable storage medium of claim 22 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to an error arising during execution of said method.
- 32-42 (Cancelled)
43. (Currently Amended) A system for modifying a software application comprising:  
~~a classfile modification system having a processor and a storage medium coupled with the processor, the classfile modification system to perform a classfile modification method, said classfile modification system to:~~  
~~modify means for inserting functions calls at entry points and exit points of each~~  
~~method associated with the software application via a bytecode modifier;~~  
~~modifying a classfile after said classfile has been compiled from a source code~~  
~~version of the software application, said classfile describing properties of a~~  
~~class within an object oriented environment; said modifying comprising:~~

~~modify means for modifying a method information structure for each method associated with the software application by adding byte code instructions to the byte code instructions of said method information structure structure's respective method, said byte code instructions to cause a plug-in module's handler method associated with a plug-in handler to execute an output function for said each method, the plug-in handler to record method information associated with methods at each entry point and exit point;~~

~~add a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions;~~

~~compile means for compiling results of the modifying of the classfile, the results including method information, the method information including a dependency hierarchical tree indicating dependency order of the methods, and a time hierarchical tree indicating chronological order of the methods; and~~

~~filter means for filtering the method information, by applying filtering parameters via a filtering module, according to user preferences and the dependency and time hierarchical trees the filtering of the method information including filtering timing data, method invocations, and other method-related information.~~

44. (Previously Presented) The system of claim 43 wherein said identities are each in a character string format.
45. (Currently Amended) The system of claim 44 wherein ~~said modifying a classfile~~ further comprises further comprising:  
means for adding a field information structure to the methods, said field information structure describing a field that is to store a numeric identifier of said class.
46. (Cancelled)
47. (Currently Amended) The system of claim 43 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to an entry point of said method being reached.